REMARKS

In the Office Action dated June 25, 2002 ("Office Action"), Claims 1-20 of the subject Application were rejected. Claims 1-20 remain pending. Claims 1, 7, 8, and 14 have be amended, and the remaining Claims remain as previously presented. In view of the amendments to the Claims and the arguments presented below, it is respectfully submitted that the Claims are in condition for allowance.

ARGUMENTS

1. The drawings are objected to because of informalities sited by the draft person. The Office Action further indicates that corrected drawings are required in reply to the Office Action to avoid abandonment of the Application. During a telephone conversation with Examiner Wood on October 24, 2002, the Examiner indicated that formal drawings to correct the cited informalities may be submitted following receipt of a Notice of Allowance in the subject case. Therefore, corrections to the cited informalities are not provided with the current response.

2. Claim 7 was rejected under 35 USC §112, second paragraph, as being indefinite. Claim 7 has been amended to depend from Claim 5 rather than from Claim 1. This provides an antecedent basis for "first selection circuit" in line 6. With this change, it is believed Claim 7 satisfies the requirements of 35 USC §112, and it is respectfully requested that this rejection be withdrawn.

3. Claims 1, 8-12, 14 and 16-20 were rejected under 35 USC §102(b) as being anticipated by U.S. Patent No. 5,325,495 to McLellan ("McLellan"). This rejection is respectfully traversed.

Before discussing the rejection in detail, the amended language of Claim 1 is considered. This language includes the following:

"a pipeline fetch circuit ... to retain a second predetermined number of instructions simultaneously, each of said second predetermined number of instructions being *in a respectively different stage of processing* within said pipeline fetch circuit..." (Claim 1 lines 9-12, emphasis added.)

Claim 1 also states that the pipeline fetch circuit is coupled directly to a pipeline execution circuit. An instruction advances within the pipeline fetch circuit independently of the advancing of instructions within the pipeline execution circuit. (Claim 1 lines 13-15.)

The pipeline fetch circuit described in Claim 1 controls the fetch stages of the pipeline shown as stages 0Y through 3Y of Figure 1. (See Applicants' Specification page 12 lines 23-24.) During each of these stages, a different type of processing activity is completed. Specifically, during the 0Y stage, address generation is performed. During the 1Y stage, an instruction is retrieved from the cache memory. Following instruction retrieval, pre-decode of the instruction being during the 2Y stages. Instruction decode continues during the 3Y stage. (See Specification page 12 lines 13-16.)

The logic associated with the fetch stages of instruction execution is shown in Figures 10 and 11. Instruction address generate logic 200 performs address generation occurring during stage 0Y. Other logic within address generation section reads an instruction from I-FLC 38 during stage 1Y. Pre-decode logic 256 performs instruction pre-decode during stage 2Y. Finally, decode logic 260 performs instruction decode during stage 3Y. (See Specification page 20 lines 20 through page 25 line 2.)

As can be understood from the description in Applicants' Specification and Drawings discussed above, each instruction retained within the pipeline fetch circuit is in a respectively different stage of processing. These stages include the address generation, instruction fetch, pre-decode, and decode stages of execution.

The Examiner states that the first sections of the McLellan pipeline shown in McLellan Figure 1 teach Applicants' pipeline fetch circuit. (Office Action page 4, line 6.) According to one interpretation of this argument, the Examiner is stating that the McLellan Pipeline stages 1 through Q correspond to Applicants' fetch circuit. Applicants' Representative respectively disagrees. Applicants' Claim 1 describes that each of the instructions retained within the pre-fetch circuit are in a *respectively different stage of processing*, as was discussed above. In contrast, McLellan discloses a system wherein the instruction within the Q-stage is not being processed at all, but instead is merely being stored. Moreover, the McLellan instructions within Stage-3 and Q-Stage will eventually be at the same state in the processing progression: they will both have completed Stage-3 and will be waiting to start Stage-4. Thus, the instructions are no in *different* stages of processing. For these reasons, McLellan does not teach Applicants' pre-fetch circuit of Claim 1.

According to another interpretation of the Examiner's argument, Applicants' fetch circuit is being equated to McLellan pipeline Stages 1 through 3, exclusive of the Q stage. However, in this case, the McLellan "fetch circuit" is not directly coupled to the McLellan pipeline execution circuit, which includes McLellan Stage 4 and remaining stages. Instead, the fetch circuit is coupled to the execution circuit through the Q stage. Thus, McLellan does not teach Applicants' amended Claim 1, which claims the pipeline fetch circuit being coupled directly to the pipeline execution circuit.

For at least the foregoing reasons, amended Claim 1 is not taught by McLellan according to either interpretation of the Examiner's argument, and this rejection should be withdrawn.

Turning now to a discussion of Claim 8, this independent Claim has been amended to include a limitation similar to that discussed above in regards to Claim 1. Specifically, Claim 8 describes a plurality of fetch logic sections wherein at least one of said plurality of fetch logic sections is coupled directly to at least one of said plurality of execution logic sections. In contrast, any fetch logic sections of McLellan are coupled to execution logic sections indirectly through the Q stage. Thus, McLellan does not teach Applicants' invention, and this rejection should be withdrawn.

Claims 9-12 depend from Claim 8, and are allowable for the reasons discussed above in regards to Claim 8. Claims 9-12 further include additional aspects of Applicants' invention that are not taught by McLellan. For example, Claim 9 includes a selection circuit to allow an instruction to be passed between first and second ones of the fetch logic sections if the second one of the sections is not executing an instruction. The Examiner states this is an inherent mechanism for handling stalls. Applicants' Representative disagrees. Many different mechanisms are available for handling stalls. For example, McLellan teaches a mechanism of inserting a Q-stage between the McLellan "pipeline fetch circuit" and the McLellan "pipeline execution circuit" to handle stalls. Thus, the additional limitation of Claim 9 is not believed to be inherent in pipeline design. For this additional reason, Claim 9 is allowable over this rejection, which should be withdrawn.

Claim 10 claims a select circuit that includes a control circuit to allow an instruction to be passed between ones of the fetch logic sections based on the availability of two of the fetch logic sections. The Examiner states that this is inherent in pipeline designs.

Applicants' Representative again disagrees. There is nothing inherent about predicating the advancement of an instruction from a first to a second logic section if the second logic section is, but the third logic section is not, executing an instruction, as is claimed by Claim 10. This aspect is not taught, or even suggested by McLellan, and Claim 10 is allowable over this rejection for this additional reason.

Claim 11 claims a select circuit having a control circuit to allow any instruction to be passed between a first and second fetch logic section if an execution logic section is performing a predetermined function. The Examiner states that this is inherent in a pipeline design, since if a pipeline stall occurs, the pipeline would not allow an instruction to progress. (Office Action page 8, second to the last paragraph.) However, Claim 11 is not merely claiming that the select circuit prevents the progression of an instruction based on a stall condition. Claim 11 is claiming a feature that controls the advancement of an instruction between two specific sections within the fetch logic based on whether a *predetermined function* is occurring with an execution logic section. This feature is certainly not inherent in pipeline designs generally, and is not taught by McLellan. For this additional reason, Claim 11 is allowable over this rejection.

Claim 12 claims a second select circuit to allow any instruction to be passed between a second and third one of the fetch logic sections if the third one of the fetch logic section is not executing an instruction. McLellan does not disclose this type of control within the "fetch logic" stages 1 through 3, nor is this type of logic necessary in a pipeline. For example, the pipeline shown in McLellan Figure 1 appears to allow stalls only in the execution stages 4 and beyond. Therefore, the type of logic claimed in Claim 12 is not necessary, or even operable, within the McLellan system. For this additional reason, Claim 12 is allowable over this rejection, which should be withdrawn.

Turning now to amended independent Claim 14, this Claim includes limitations similar to those discussed above with respect to Claims 1 and 8. Specifically, step a.) includes providing at least one of the fetch logic sections that is directly coupled to an execution logic section. As discussed above, at most, McLellan discloses connecting fetch logic to execution logic through the Q stage, as discussed by the Examiner. (Office Action, page 10, line 8.) Thus McLellan does not teach Applicants' method of Claim 14, and for at least this reason, this rejection should be withdrawn.

Claim 16 depends from Claim 14 and is allowable over this rejection for at least the reasons discussed above in regards to Claim 14. Claim 16 further includes the step of allowing an execution logic section to retain an instruction longer in order to retrieve an operand. The Examiner states that although this is not shown by McLellan, it is inherent in pipeline design. Applicants' Representative disagrees. For example, in a system having a cache that is large enough to prevent cache misses for the applications being executed, no stall is ever created because of failure to retrieve an operand. Therefore, the pipeline need not perform the type of step discussed in Claim 16, which is not necessarily "inherent" in pipeline design. For this additional reason, Claim 16 is allowable over this rejection.

Claims 17 and 18 depend from Claim 14, and are allowable over this rejection for at least the reasons discussed above with respect to Claim 14.

Claim 19 depends from Claim 14, and includes the additional aspect of repeating a fetching step for a predetermined number of instructions irrespective of whether any of the fetch logic sections or the execution logic sections begins processing another instruction. This refers to the aspect of Applicants' invention wherein the instruction address generation section 111 (Figure 10) retrieves instructions for storage in instruction queue 218 before the pre-decoding or decoding begins. (See Applicants' Specification, pages 21 and 22.) The Examiner states that this step is inherent in the design of the McLellan pipeline. (Office Action page 13 lines 9-11.) Applicants' Representative disagrees. The McLellan Stage-1, which performs instruction fetching, does not appear to perform this function, nor is this function a requirement to implement the pipeline design of McLellan. McLellan does not teach, or even suggest, this additional aspect of Applicants' invention, and Claim 19 is therefore allowable over this rejection for this additional reason.

Claim 20 depends from Claim 14 and is allowable over this rejection for at least the reasons discussed above in regards to this Claim.

4. Claims 2-6 were rejected under 35 USC §103(a) a being unpatentable over McLellan in view of U.S. Patent No. 5,778,423 to Sites et al. ("Sites"). This rejection is respectfully traversed.

Claims 2-6 depend from Claim 1. Sites adds nothing to McLellan that would teach or suggest the limitations of Applicants' Claim 1. Therefore, Claims 2-6 are allowable over this rejection for at least the reasons discussed above in reference to Claim 1. These Claims further include additional scopes and aspects not taught or suggested by the cited

combination of references and are further allowable over this rejection for this additional reason.

5. Claims 7, 13, and 15 were rejected under 35 USC §103(a) a being unpatentable over McLellan in view of U.S. Patent No. 5,577,259 to Alferness et al. ("Alferness"). This rejection is respectfully traversed.

Claim 7 depends from Claim 1. Alferness adds nothing to McLellan that would teach or suggest the limitations of Applicants' Claim 1. Therefore, Claim 7 is allowable over this rejection for at least the reasons discussed above in reference to Claim 1. Additionally, Claim 7 includes the limitation of a pipeline execution circuit that includes a microcode-controlled sequencer to control execution of extended stages of execution for extended-mode instructions. The first selection circuit includes a control circuit to allow an instruction to enter the pre-decode stage of processing while the extended-mode instruction are not advancing within the pipeline execution circuit. The Examiner states that the additional limitations are shown by Alferness. In particular, the Examiner states that column 4 lines 63-67 discloses Applicants' control circuit to allow an instruction to enter the pre-decode stage of processing while the microcode-controlled instructions are not advancing within the pipeline. Applicants' Representative disagrees. While Alferness discloses a microcode-controlled sequencer generally, Alferness does not disclose the control circuit. The cited Alferness passage states:

> "The microcode controller, however, does not need to respond to these conditions, since the overlapping of machine instruction execution in the pipeline is suspended during execution of an extended-cycle instruction".

This passage is discussing the very limitation discussed in Applications' Specification in reference to prior art systems like Alferness. Specifically, in such prior art systems, the fetch stages of a pipeline are suspended when non-standard (microcode-controlled) instructions are executed. (Applicants' Specification page 4 lines 8-19.) This prevents other instructions from entering the fetch portion of the pipeline, thereby diminishing throughput. The cited Alferness passage is therefore describing the very problem the system of Claim 7 is designed to prevent. In teaching the prior art design discussed in Applicants' Specification, Alferness actually *teaches away* from Applicants' invention of Claim 7.

To summarize, nothing in the cited combination of references teaches or suggests the invention of Claim 7. For this additional reason, Claim 7 is allowable over this rejection, which should be withdrawn.

Claim 13 depends from Claim 8. Alferness adds nothing to McLellan that would teach or suggest the limitations of Applicants' Claim 8. Therefore, Claim 13 is allowable over this rejection for at least the reasons discussed above in reference to Claim 8. Additionally, Claim 13 includes the limitation involving the de-coupling of the fetch and the execution stages during microcode control, as is discussed above in reference to Claim 7. This limitation is not taught or suggested by Alferness. In fact, Alferness actually teaches away from the invention of Claim 13. For this additional reason, Claim 13 is allowable over this rejection, which should be withdrawn.

Claim 15 depends from Claim 14. Alferness adds nothing to McLellan that would teach or suggest the limitations of Applicants' Claim 14, and Claim 15 is allowable over this rejection for at least the reasons discussed above in reference to Claim 14. Additionally, Claim 15 includes limitations similar to those discussed above in reference to Claims 7 and13. These limitations are not taught or suggested by Alferness, which teaches away from the invention of Claim 15. For this additional reason, Claim 15 is allowable over this rejection, which should be withdrawn.

6. The prior art of record that is not relied upon has been reviewed and is considered to be of general interest only. Applicants' Representative does not necessarily agree with the significance the Examiner assigns to these references.

Conclusion

In the Office Action dated June 25, 2002, Claims 1-20 of the subject Application were rejected. Claims 1-20 remain pending. Claims 1, 7, 8, and 14 have be amended, and the remaining Claims remain as previously presented. In view of the amendments to the Claims and the arguments presented above, it is submitted that the Claims are in condition for allowance, and an early Notice of Allowance is respectfully requested.

Respectfully submitted,

Beth L McMahon 10/25/2002

Beth L. McMahon
Attorney for Applicants
Reg. No. 41,987
Tele No. (651) 635-7893

Unisys Corporation
M.S. 4773
P.O. Box 64942
St. Paul, MN 55164-0942

BLM/clk

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS:

Claim 1 (Once Amended):

Claim 1: For use in an instruction processor that executes instructions included in a predetermined instruction set at an execution rate determined by a system clock signal, a synchronous instruction pipeline, comprising:

a pipeline execution circuit to process a first predetermined number of instructions simultaneously, each of said first predetermined number of instructions being in a respectively different stage of execution within said pipeline execution circuit, instructions being capable of advancing to a next stage of execution within said pipeline execution circuit at a time determined by the system clock signal; and

a pipeline fetch circuit coupled directly to said pipeline execution circuit to [process] retain a second predetermined number of instructions simultaneously, each of said second predetermined number of instructions being in a respectively different stage of processing within said pipeline fetch circuit, an instruction being capable of advancing to a next stage of execution within said pipeline fetch circuit at a time determined by the system clock signal and independently of the times at which instructions advance to a next stage of execution within said pipeline execution circuit.

Claim 7 (Once Amended):

7. The synchronous instruction pipeline of Claim [1] 5, wherein said pipeline execution circuit includes a microcode-controlled sequencer to control execution of extended stages of execution of extended-mode ones of the instructions, wherein during said extended stages of execution, ones of the instructions being executed by said pipeline execution circuit are not advancing to a next stage of execution within said pipeline execution circuit, and wherein said first selection circuit includes a control circuit to allow an instruction to enter said pre-decode stage of processing while said extended-mode ones of the instructions are not advancing to a next stage of execution within said pipeline execution circuit.

Claim 8 (Once Amended):

8. For use in an instruction processor that executes instructions of a machine instruction set, a synchronous pipeline system comprising:

a plurality of execution logic sections, each of said execution logic sections being coupled to at least one respective other one of said execution logic sections, each of said execution logic sections to perform a predetermined stage of execution of any of the instructions, and whereby each of said execution logic sections is capable of receiving a new instruction to process at predetermined time increments; and

a plurality of fetch logic sections wherein at least one of said plurality of fetch logic sections is coupled directly to at least one of said plurality of execution logic sections, each of said fetch logic sections being coupled to at least one respective other one of said fetch logic sections, each of said fetch logic sections to perform a predetermined pre-execution stage of instruction execution, each of said fetch logic sections being capable of receiving a new instruction to process at said predetermined time increments and in a manner that is independent of whether any of said plurality of execution logic sections receives a new instruction to process.

Claim 14 (Once Amended):

Claim 14: For use in an instruction processor having a synchronous instruction pipeline that executes instructions at a rate determined by a system clock, the instruction pipeline including a predetermined number of execution logic sections coupled to each other in sequence, each to perform a respectively different stage of execution on any instruction, and a predetermined number of fetch logic sections coupled to each other in sequence, each to perform a respectively different stage of pre-execution on any instruction, [and wherein at least one of the fetch logic sections is coupled to at least one of the execution logic sections,] a method of processing instructions, comprising the steps of:

(a) providing at least one of the fetch logic sections that is coupled directly to at least one of the execution logic section;

[(a)](b) processing a respective one of the instructions by each of the execution logic sections for a first predetermined time period;

[(b)](c) allowing ones of the execution logic sections to each pass said respective one of the instruction to another coupled one of the execution logic sections after said first

predetermined time period elapses;

[(c)](d) allowing at least one of the execution logic sections to retain said respective instruction for longer than said first predetermined time period; and

[(d)](e) allowing ones of the fetch logic sections each to begin processing a respective instruction during a subsequent predetermined time period that is subsequent to said first predetermined time period if said each of the fetch logic sections was not processing a respective instruction during said first predetermined time period.